Claim Amendments

Claims 1-15 (Cancelled)

16. (Currently amended) A method for mapping a texture onto a surface of a computer generated object represented by a plurality of pixels, comprising ~~the steps of~~:

dividing a texture map into blocks, the texture map comprising a plurality of texels, each texel having an associated value;

determining two block values for each block, which block values are representative of the values of texels in the block;

compressing the texture map by assigning to each texel one of the two block values associated with the block of which it is part; and

mapping said compressed texture map onto the surface of the computer generated object.

B4

17. (Currently amended) [[A]] The method as set forth in claim 16, wherein the block values associated with the texture map are quantized to a smaller number of bits.

18. (Currently amended) [[A]] The method as set forth in claim 16, wherein ~~the step of~~ determining two block values for each block further comprises:

calculating a tensor of inertia from texel values;

determining an eigenvector having a smallest eigenvalue from said tensor;

multiplying said smallest eigenvalue eigenvector with said texel values; and

splitting the texel values in two groups by comparing a result of said multiplication with a threshold value.

19. (Currently amended) [[A]] The method as set forth in claim 16, wherein the texture map corresponds to a filtered texture map of lesser detail than a texture map of full detail.

20 (Currently amended) [[A]] The method as set forth in claim 16, wherein ~~the step of~~ mapping said compressed texture map onto the surface of the computer generated object further comprises:

for each pixel which represents the computer generated object, accessing said compressed texture map at least one time; and

responding to said compressed texture map being accessed more than one time by interpolating results of the accesses.

21. (Currently amended) [[A]] The method as set forth in claim ~~20~~ 16, wherein ~~the step of~~ mapping said compressed texture map onto the surface of the computer generated object further comprises:

approximating true pixel color by performing a number of texturing operations according to a geometric shape of a projection of a pixel on the texture and averaging results of said texturing operations.

22. (Currently amended) [[A]] The method as set forth in claim ~~21~~ 16 wherein the texture is an environment map.

23. (Currently amended) [[A]] The method as set forth in claim ~~22~~16 wherein ~~at least one of said texture mapping, environment mapping, reflectance mapping and detail~~ the mapping is carried out in real time using dedicated arithmetic units.

24. (Currently amended) A device for ~~at least one of texture mapping, environment mapping, reflectance mapping and detail~~ mapping comprising:

means for compressing a texture map using blockwise two-level (one bit) quantization of brightness values or colors;

means for storing said compressed texture map on a storage medium;

means for mapping said stored texture map onto the surface of the computer generated object;

dedicated arithmetic unit means; and

memory units for storing ~~at least one of texture, environment, reflectance and detail~~ maps.

Claims 25-28 (Cancelled)

29. (Currently amended) [[A]] The method as set forth in claim ~~21~~ 16 wherein the texture is a reflectance map.

30. (Currently amended) [[A]] The method as set forth in claim ~~21~~ 16 wherein the texture is a detail map.

31. (Currently amended) The ~~texturing unit of~~ method as set forth in claim 16, wherein each block value represents ~~the~~ luminance of a texel.

32. (Currently amended) The ~~texturing unit of~~ method as set forth in claim 16, wherein each block value represents an index into a look-up table.

33. (Currently amended) The ~~texturing unit of~~ method as set forth in claim 16, wherein each block value represents ~~the~~ color of a texel.

34 (Cancelled)

35. (Original) A texturing unit for mapping a texture to a surface of a computer generated object, which texture comprises a plurality of blocks, each block comprising a plurality of texels and having two block values associated with the block, and each texel of each block corresponding to one of the two block values associated with the block, the texturing unit comprising:

a Random Access Memory (RAM) for storing the two block values associated with each block of the texture and a value for each texel, which value indicates the block value to which the texel corresponds;

a decompression unit coupled to the RAM, for accepting from the RAM values representing eight texels and the block values associated with each block of which the eight texels are part, and for determining eight decompressed texel values therefrom;

a trilinear interpolator coupled to the decompression unit, for accepting from the decompression unit the eight decompressed texel values and interpolating an interpolated value therefrom; and

an output port coupled to the trilinear interpolator, for transmitting the new value to a device coupled to the output port.

36. (Original) The texturing unit of claim 35, wherein the RAM is configured such that values for eight texels can be accessed substantially simultaneously, the eight texels comprising four texels from a first level and four texels from a second level, where the first level is one level higher than the second level.

37. (Original) The texturing unit of claim 36, wherein the four texels from the first level represent a two-by-two block of contiguous texels within the first level of the mipmap, and the four texels form the second level represent a two-by-two block of contiguous texels within the second level of the mipmap.

38. (Currently amended) The texturing unit of claim ~~36~~ 35, wherein each decompressed texel value represents an index into a look-up table.

39. (Currently amended) The texturing unit of claim ~~36~~ 35, wherein each decompressed texel value represents the color of a texel.

40. (Original) The texturing unit of claim 35, wherein the RAM, the interpolator, and the output port are part of a single chip.

41. (Original) The texturing unit of claim 35, wherein the interpolator comprises at least one dedicated arithmetic unit.

42. (Cancelled)

43. (Cancelled)

B4

44. (Cancelled)

45. (Original) The texturing unit of claim 35, wherein the texture comprises a plurality of blocks, each block comprising a plurality of texels and having two block values associated with the block, and each texel of each block corresponding to one of the two block values associated with the block, the information stored in the RAM comprising:

the two block values associated with each block of the texture; and

a value for each texel, which value indicates the block value to which the texel corresponds.

46. (Currently amended) The texturing unit of claim 35, wherein each texel value represents ~~the~~ luminance of a texel.

47. (Original) The texturing unit of claim 35, wherein each texel value represents an index into a look-up table.

48. (Currently amended) The texturing unit of claim 35, wherein each texel value represents ~~the~~ color of a texel.

49. (Currently amended) The texturing unit of claim 35, wherein each decompressed texel value represents ~~the~~ luminance of a texel.

50. (Currently amended) The texturing unit of claim ~~36~~ 35, wherein the texture is a view of an environment of a scene.

51. (Currently amended) The texturing unit of claim ~~36~~ 35, wherein the texture is a reflectance map, and the texel values are specular reflectance coefficients.

52. (Currently amended) The texturing unit of claim ~~36~~ 35, wherein each texel is associated with a horizontal detail offset and a vertical detail offset, which horizontal detail offset and vertical detail offset are pointers into a detail map associated with the texture, which detail map is stored in the RAM.

53. (Original) The texturing unit of claim 52, wherein the detail map is a mipmap.

54-57 (cancelled)

58

54. (New) A method for mapping a texture onto a surface of a computer generated object represented by a plurality of pixels, comprising:

dividing a texture map into blocks, the texture map comprising a plurality of texels, each texel having an associated value;

determining two block values representative of the texel value for each block, the determining comprising calculating a tensor of inertia from the texel values, determining an eigenvector having a smallest eigenvalue from the tensor, multiplying the smallest eigenvalue eigenvector with the texel values; and splitting the texel values in two groups by comparing a result of the multiplication with a threshold value; and

compressing the texture map by assigning to each texel one of the block values associated with the block of which it is part.